A Skeleton Based Descriptor for Detecting Text in Real Scene Images

Mehdi Felhi
Université de Lorraine-
LORIA UMR 7503/
Océ Print Logic Technologies
mehdi.felhi@loria.fr

Nicolas Bonnier
Océ Print Logic Technologies
nicolas.bonnier@oce.com

Salvatore Tabbone
Université de Lorraine-
LORIA UMR 7503
tabbone@loria.fr

Abstract

In this paper, we present a new method for text extraction in real scene images. We propose first a skeleton based descriptor to describe the strokes of the text candidates that compose a spatial relation graph. We then apply the graph cuts algorithm to label the nodes of the graph as text or non-text. We finally refine the resulted text lines candidates by classifying them using a kernel SVM. To validate this approach we perform a set of tests on the public datasets ICDAR 2003 and 2011.

1. Introduction

Text detection methods could be categorized in two main classes; geometrical based and statistical based methods. Geometrical based approaches try to describe the text with a priori knowledge about the text such as its color, local texture, edge and/or solidity. For instance, edge-based methods such as the work of Sato et al. [13] suppose that the text have high contrast with the background and therefore look for the edgy regions of the image and characterize the text by means of some geometrical assumptions. Color clustering is also frequently used to extract text. In [8] the authors propose a method that scans the image horizontally and cluster lines that belong to the same color class, and then they consider these clusters to be the text characters. Unfortunately, geometrical based methods could fail when the text does not present a high contrast with its background and when some of the assumptions (a priori knowledge) are not respected in the image. Statistical based methods include probabilistic models and sparse representation. Probabilistic models such as the Markov Random Field model and the Conditional Random Field model (CRF) [6] are well used to extract text from complex background. These models combine the unary properties of the text (such as the compactness and height/width ratio of text characters) with the binary or contextual properties (similarity and spacing between the text line components). Sparse representation is also one of the most common used approaches for describing the text. In [12], authors use an over-complete dictionary for text signals trained using the K-SVD algorithm in order to label the connected components as text or non-text regions.

Our work could be considered as a connected component approach and more precisely as a stroke based approach that extends the existing methods [1, 2]. In general, text strokes present a uniform width and color. Several papers exploited this property. For instance, in [2] authors introduced the Stroke Width Transform (SWT), this transform permits to determine the width of each part of the connected component and then select those that have low variations (below a certain threshold) and consider them as text character candidates. The SWT is calculated by determining the distance separating the two opposite gradients along the edges of the connected component. This method performs efficiently good detection rates when the text is well contrasted. Alternatively, the gradient along the character's edge is not well determined and the calculation of the stroke width might fail. H. Chen et al. [1] proposed a new version of stroke width transform based on the distance function that overcomes the aforementioned problem. Formally, the authors [1] calculate the distance that separate each pixel in the considered connected component from the background using the distance transform, and then spread the maximum distances along the stroke width. Finally, they consider that the real width of the stroke is twice the resulted transform. This method extracts the text candidates in the grayscale level by using the MSER, and then prunes them by means of Canny edges. However, Canny edge detector might fail to detect edges of the text characters if the text is not highly contrasted in the grayscale level. Moreover, most of the process done in this work is local and do not take

into account global information which is crucial. In this paper, we propose a global approach that aims to overcome the above mentioned problems and improve the performances of the text detection.

2. Proposed approach

The proposed approach consists of four main steps:
1- Pre-processing step: we extract the text candidates using the Maximally Stable Extremal Regions (MSER).
2- Graph construction: by eliminating connected components that are most likely non-text and connecting similar and neighboring components.
3- Processing step: we apply our text descriptor on each node of the graph. We eliminate nodes that have very low probability to be text. At the end of this step, each node of the graph is labeled as text or non-text using the Graph Cuts.
4- Refinements: this step aims to eliminate false positives.

2.1. Preprocessing

We use the Maximally Stable Extremal Regions (MSER) algorithm [9] to extract connected components. This technique is widely used as a blob detection method. Basically, an MSER region is a region that is either brighter or darker than its neighboring background. This method demonstrated its superiority comparing to several existing texture patch descriptors [4] in terms of robustness and efficiency. Besides, the MSER has been used as a preprocessing step for the text detection method [1] and authors have shown its ability to easily detect text characters. After the MSER extraction step, we calculate for each component C the following geometrical properties in order to characterize it: 1) The mean value μ^C: This value corresponds to the center of the region C : $\mu^C = \frac{1}{|C|}\sum_{i\in C}[x_i, y_i]$ where $|C|$ denotes the number of pixels in C. 2) The parameter a_c (respectively b_c) that measures the length of the major (respectively minor) axis of the ellipse fitted to the region C. The parameters of this ellipse are inferred from the normalized second central moments corresponding to the region C (we used the VLFeat open source library [15] for calculating these parameters). In this paper we call a_c (respectively b_c) the major (respectively the minor) elongation of the region C. We choose these parameters for two main reasons; first, the a_c/b_c ratio approximates the height/width ratio of the connected component regardless its orientation; this approximation gives much better precision than the bounding box technique when text characters are skewed. Second, all the text characters that belong to the same text line have similar major elongations. This property will be exploited to construct the graph of text candidates.

2.2. Graph construction

The aim of this step is to construct a graph that includes the text region candidates in the image. For this purpose, we begin by applying some rules in order to eliminate the non-text regions; for instance, a very small connected component is considered as non-text, likewise, a region that has a very high height/width ratio is disregarded. Similarly, we eliminate the components that include a high number of holes. By achieving this step, we obtain the text candidate regions. We consider them as the nodes of the graph. Then, we measure the relationship between nodes in order to cluster them by computing similarities between nodes. The edges of the graph should connect only similar and neighboring nodes. For that reason, we propose three binary features that will quantify the similarity and the neighboring of two text candidate components c_i and c_j (see Table 1). The proposed features are chosen in such a way that they are independent to the scale and the orientation of the text:
- Relative spatial distance: Since text components belonging to the same text line are close. Moreover, neighboring nodes are likely to have the same label (text or non-text).
- Scale ratio: We compare the lengths of major elongations of the text candidates. In fact, text characters that belong to the same text line have close major elongations contrary to their minor: for instance, the character "i" presents very low minor elongation length comparing to the character "m". However, the two characters have close major elongation lengths.
- Color difference: Since neighboring text characters have generally the same color.
As discussed earlier, we aim to construct a graph that links only similar components in order to cluster them in text lines candidates. For that purpose, we eliminate non-horizontal connections in the graph. Then, we classify the edges into valid and non valid connections by training an SVM classifier to two different set of couple of nodes; the first, consists of neighboring text characters belonging to a same text line, and the second is composed of a random set of neighboring non-text regions. The features RSD, SR, and CD are used as an input to the SVM.

2.3. Processing

2.3.1. Text descriptor Our descriptor is based on skeleton and distance transforms. The skeleton should describe fairly the structure of the corresponding

Feature	Definition
Relative Spatial Distance	$RSD(c_i, c_j) = \frac{\|\mu_i - \mu_j\|_2}{b_{c_i} + b_{c_j}}$
Scale Ratio	$SR(c_i, c_j) = \frac{\left\|a_{c_i} - b_{c_j}\right\|}{a_{c_i} + b_{c_j}}$
Color Difference	$\|I(c_i) - I(c_j)\|_2$

Table 1: The binary features for graph construction

component (text candidate). Thus, we need to prune the skeletal branches and eliminate non useful ones or those that make small contribution to the description of its structure. The idea behind pruning the skeletal branches is based on the fact that branches pointing toward the boundary are unnecessary ones [5]. Furthermore, text skeletal branches do not present high local stroke width variations. Therefore, we propose the following criterion to describe the CCs: the branches that present high distance variation (the distance that separate the skeleton pixels from the background) are removed. Mathematically, each skeletal branch b that presents a V_b value higher than a threshold t is eliminated. V_b is defined as following:

$$V_b = \frac{\max_{i \in b}\left(D^C(i)\right) - min_{i \in b}\left(D^C(i)\right)}{length(b)}$$

$D^C(i)$ defines the distance that separates the pixel i from the background of the component C. Throughout this paper, let sk^C denote the remainder parts of the component C skeleton. In this paper we assume that text regions present high proportion of low variation stroke width branches. To quantify this property we define the features P_c and SWV_c as following:

$$P_c = \frac{\left||C| - 2\sum_{i \in sk^C} D^C(i)\right|}{\left||C| + 2\sum_{i \in sk^C} D^C(i)\right|}, SWV_c = \frac{std\left(\left\{D^C i\right\}_{i \in sk^C}\right)}{mean\left(\{D^C i\}_{i \in sk^C}\right)}$$

Note that P_c is varying in the interval $[0, 1]$. The lower the P_c value, the more-likely C presents a text character. Hence we propose to use a monotonically decreasing function of this parameter as a probability for a component to be text. By doing a statistical analysis of a set of text characters with their corresponding P_c values, we remarked that the resulted histogram of a component being a text could be approximated by a folded normal distribution [11]. We approximated this distribution by estimating its variance using the classical unbiased estimator. As a conclusion, the probability density function f of a component C being text could be written as follows: $f(P_c) = \frac{2}{\sqrt{2\pi\hat{\sigma}}} exp\left(\frac{-P_c^2}{2\hat{\sigma}^2}\right)$ where, $\hat{\sigma}$ is the estimated variance of the folded distribution. The second feature SWV_c defines the global stroke variation of the connected component C. This feature will be exploited in the refinement step.

2.3.2. Graph labelling The image segmentation is insured by means of the graph cuts algorithm. The entry of the graph cuts is the graph described in the previous paragraphs where the probability of each node C being text is equal to $f(P_C)$ while its probability to be non-text is equal to $f(0) - f(P_C)$. The main advantage of the graph cuts stage is to get similar labels for neighboring nodes by means of a smoothing function S.

3. Classification and refinements

This step aims to increase the precision rates of the proposed approach by eliminating false positives. For this purpose, we define a feature vector $F(T)$ that describes each text line T. This vector includes the four following features: 1) The mean value of the height/width ratio throughout the text components that belong to T. 2) The mean solidity value. 3) The mean value of the stroke width variation SWV. 4) The mean value of the gradient variation along the components contours: we introduce this feature because text lines are often surrounded by a uniform background contrary to many other objects that could be detected in a real scene image. The gradient variation along a component C is defined as follows:

$$GV^C = \frac{std\left(\{g(i)\}_{i \in B^C}\right)}{mean\left(\{g(i)\}_{i \in B^C}\right)} \quad (1)$$

where g is the gradient function of the image I and B^C denotes the boundary pixels of the connected component C. We trained a Gaussian Kernel SVM on a set of text line candidates in order to separate text from non-text lines by using their corresponding feature vectors F. Some examples of text detection generated by our approach are shown in Fig. 1.

4. Experimentations

We evaluated our new approach on the public datasets ICDAR 2003 [7] and ICDAR 2011 [14] since many text detection works were evaluated on them. These two datasets were designed for text locating competitions. Each dataset consists of two sets of images; the first set is devoted for training and the second one is devoted for test. All the images of the ICDAR datasets are natural scene ones and were captured using a digital camera under different luminosity conditions. Mainly, the two classical measures are used for evaluation; precision and recall rates. The mathematical formulas of

Figure 1: The result of text detection generated by the proposed method on two sample images from the ICDAR 2011 dataset [14]: On the left, original images. On the right, corresponding segmented images using our approach

Method	Proposed approach	Epshtein [2]	Chen [1]	Minetto [10]	Fabrizio [3]
Precision	**0.75**	0.73	0.73	0.63	0.46
Recall	**0.61**	0.60	0.60	**0.61**	0.39

Table 2: Experimental results on ICDAR 2003 dataset

these measures are explicitly defined in [7] for ICDAR 2003 dataset evaluation. The evaluation method proposed in [16] was employed to evaluate our scene text detection approach on the ICDAR 2011 dataset. Note that [16] was designed to overcome the problem of over-, under- and missed segmentation. We compared the performances of our work with several existing text detection methods (with [2, 1, 10, 3] on the ICDAR 2003 dataset and with the works that participated in the ICDAR 2011 competition [14] (See Table 2 and Table 3).

5. Conclusion

We have presented in this paper a novel stroke based approach for text detection in real scenes images. The method begins with clustering similar and valid neighboring components in a graph. Then, we introduced a new text descriptor as an attribute for each node of the graph. This descriptor is defined as the proportion of low variation stroke width branches that compose the components' skeletons. After that, each node of the constructed graph is labeled as text or non-text due to a graph cuts process. Finally the image is segmented after a classification step that aims to refine the results. We have shown that our approach performs good results comparing to existing methods. Further works will consist in extending our detection approach to the multi-orientation case since our text features are independent to the orientation.

Method	Proposed approach	Kim's Method	Yi's Method	TH-TextLoc System
Precision	**0.84**	0.83	0.67	0.67
Recall	0.60	**0.62**	0.58	0.58

Table 3: Experimental results on ICDAR 2011 dataset

References

[1] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. *ICIP*, pages 2609–2612, 2011.

[2] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. *CVPR*, pages 2963–2970, 2010.

[3] J. Fabrizio, M. Cord, and B. Marcotegui. Text extraction from street level images. *CMRT*, page 199204, 2009.

[4] P.-E. Forssén and D. G. Lowe. Shape descriptors for maximally stable extremal regions. *ICCV*, pages 1–8, 2007.

[5] J.-H. Jang and K.-S. Hong. Detection of curvilinear structures and reconstruction of their regions in gray-scale images. *Pattern Recognition*, 35(4):807–824, 2002.

[6] M. Li, M. Bai, C. Wang, and B. Xiao. Conditional random field for text segmentation from images with complex background. *Pattern Recognition Letters*, 31(14):2295–2308, 2010.

[7] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. Icdar 2003 robust reading competitions. *ICDAR*, pages 682–687, 2003.

[8] V. Y. Mariano and R. Kasturi. Locating uniform-colored text in video frames. *ICPR*, pages 4539–4542, 2000.

[9] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *BMVC*, 2002.

[10] R. Minetto, N. Thome, M. Cord, J. Fabrizio, and B. Marcotegui. Snoopertext: A multiresolution system for text detection in complex visual scenes. *ICIP*, pages 3861–3864, 2010.

[11] L. Nelson. The folded normal distribution. *Journal of Quality Technology*, 12(4):236–238, 1980.

[12] W. Pan, T. D. Bui, and C. Y. Suen. Text detection from scene images using sparse representation. *ICPR*, pages 1–5, 2008.

[13] T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith. Video ocr for digital news archive. *CAIVD*, pages 52–60, 1998.

[14] A. Shahab, F. Shafait, and A. Dengel. Icdar 2011 robust reading competition challenge 2: Reading text in scene images. *ICDAR*, pages 1491–1496, 2011.

[15] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

[16] C. Wolf and J.-M. Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *IJDAR*, 8(4), 2006.